Presented at 50th Annual Air Traffic Control Conference, October 31, 2005

Using Graphical Software for Evaluating Aircraft Trajectory Predictions

C. Santiago III
The Titan Corporation
confesor.ctr.Santiago@faa.gov

M. M. Paglione
FAA William J. Hughes Technical Center
mike.paglione@faa.gov

A. S. Rusu
Rowan University
rusu@rowan.edu

S. Liu
General Dynamics Network Systems
shurong.ctr.liu@faa.gov

R. D. Oaks
General Dynamics Network Systems
robert.ctr.oaks@faa.gov

S. Putney
Joseph Sheairs Associates, Inc.
seth.ctr.Putney@faa.gov

***Confesor Santiago** is a Cooperative Education Student with The Titan Corporation and a graduate student at Rowan University, Glassboro, NJ. **Mike M. Paglione** is the Conflict Probe Assessment Team Lead in the Simulation & Analysis Group with the Federal Aviation Administration. **Dr. Adrian S. Rusu** is an Assistant Professor at Rowan University; and established the Software Engineering, Graphics, and Visualization (SEGV) Research Group at Rowan University. **Shurong Liu** is a Senior Staff Member with General Dynamics Network System/Signal Solutions. **Robert D. Oaks** is a Principal Staff Member with General Dynamic Network Systems/Signal Solutions. **Seth Putney** is a Senior Systems Engineer with Joseph Sheairs Associates, Inc.*

Abstract

The Federal Aviation Administration (FAA) air traffic control system uses Decision Support Tools (DSTs) such as the User Request Evaluation Tool (URET) and the Traffic Management Advisor (TMA) to assist air traffic controllers to separate air traffic. DSTs predict aircraft flight paths and foretell potential conflicts. The accuracy of the trajectories generated by the DSTs determines their overall performance. The Conflict Probe Assessment Team, located at the FAA William J. Hughes Technical Center in New Jersey, has measured the accuracy of the TMA and URET trajectory modelers. And they will continue to study the accuracy of trajectory modelers in planned systems. A methodology has been developed to measure the accuracy of trajectories. Various algorithms and relational databases have been fashioned to investigate and calculate the performance accuracy of trajectory prediction. This paper presents a new graphical tool called the Trajectory Graphical User Interface (TrajectoryGUI), which was jointly developed by the Rowan University's Software Engineering, Graphics, and Visualization Research Group and the FAA, to assist the investigators in measuring the accuracy. The tool provides assistance through an easy to use graphical user interface. The key features of the tool are its ability to graphically plot from a plethora of different flights and trajectories, view the plotted path, both in horizontal and vertical dimensions, zoom in

and out, apply time tags for comparisons, and export plots and statistical analysis results of the investigation for external presentation. The development of this tool has resulted in a thorough process of evaluating the trajectory prediction accuracy of DSTs, It supplements the existing data analysis environment to a visualization analysis environment. An analyst can use TrajectoryGUI to package the trajectory accuracy results and illustrate the inaccuracy.

Background

In 1996, the Federal Aviation Administration (FAA) established the Conflict Probe Assessment Team (CPAT) at the William J. Hughes Technical Center in New Jersey to evaluate the accuracy of the conflict probes in Decision Support Tools (DSTs). Since its creation CPAT has:

- measured the conflict prediction accuracy of URET [1],
- measured the trajectory modeling accuracy of both URET and CTAS [2],
- conducted a study that measured the sensitivity of the URET conflict probe to weather forecast errors [3], and
- assisted in the formal accuracy testing of URET Current Capability Limited Deployment (CCLD), which was the initial operational implementation of URET.

As a part of these analyses, CPAT has always valued tools that supplement their data analysis with visual representations of the data. For example in 2001, CPAT developed a Java application called the Proof Encounter Preparation Program (PREPS) that provides input files for an off-the-shelf product called Proof Animation[i], which animates conflict encounter data [4]. In 2003, CPAT had a summer intern supported through the National Association for Equal Opportunity in Higher Education[ii] develop a prototype of the TrajectoryGUI. This prototype application is presented in [5].

In 2004, CPAT teamed with the Software Engineering, Graphics, and Visualization (SEGV) Research Group that was established by the Department of Computer Science at Rowan University and currently has 13 Alumni, two External Collaborators, seven Student Members, and one Senior Member. Since SEGV's research is focused on Software Engineering Education, Web Browsing and Visualization, Sound Visualization, and Graphical Software and Applications, CPAT saw that collaboration with them would provide an excellent opportunity for the continuing development and enhancement of the TrajectoryGUI application, as well as enrich the academic opportunities of the participating Rowan University students and staff.

TrajectoryGUI

In 2005, the SEGV developed an upgraded version of the TrajectoryGUI for CPAT initially as part of a senior class project and continued developed through a funded internship by the FAA and a selected Rowan student. Like the prototype version developed 2001, the upgraded TrajectoryGUI is written in Java and interacts with an Oracle relational database using the Java Database Connectivity (JDBC) Application Program Interface (API)[iii] to execute the database queries for collecting flight data and flight trajectories. For this upgrade, the SEGV completely redesigned the application to use the JOGL[iv] library, which implements the OpenGL graphics standard[v], to provide the plots.

When an analyst launches Trajectory GUI the application presents a selection window, and example of which is presented in Figure 1. The analyst uses this window to select the specific data to be used for plots and tabular display during the session.

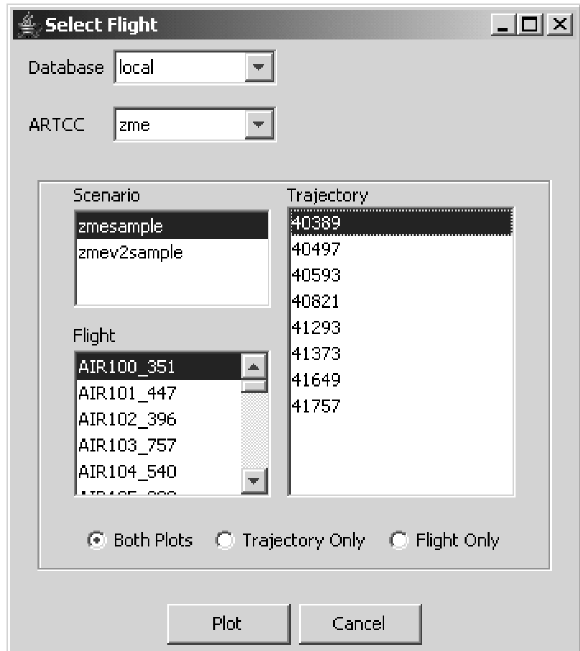

Figure 1: Selection Window

The analyst first identifies the database that contains the tables providing the desired data for this session. These databases are the databases used by CPAT to calculate trajectory accuracy measurements [6]. The available database mappings are identified and selected from a drop-down list that is maintained in a configuration file that can be edited by the analyst. The analyst then selects the appropriate Air Route Traffic Control Center (ARTCC) from another drop-down list. In the example shown in Figure 1, the analyst has identified the local database and has selected **ZME**, which is the identifier for the Memphis ARTCC.

TrajectoryGUI uses the database and ARTCC information to query the database and fill the Scenario list area, which will identify the available scenario cases. In the example, two scenarios were identified and the analyst has selected the scenario case identified as **ZMESAMPLE**. This selection causes the Flight list area to be filled, identifying the available flights. In the example, the analyst has selected flight **AIR100_351**, which represents a flight with the aircraft identification (ACID) of AIR100 and a computer identification (CID) of 351. This selection causes the Trajectory text area to be populated with the build times of the trajectories that were generated by the DST. The analyst now selects the desired trajectory to be plotted. In the example, the analyst has selected the trajectory with a build time of **40389**. At this point the analyst can use the radio buttons to select the plotting option: trajectory and flight path, trajectory only, or flight only. After the desired option has been chosen, the analyst then clicks the plot button.

TrajectoryGUI queries the database and presents the application's main window, which contains an **X-Y** plot, a **T-Z** plot, and the **Metrics** table for the selected flight. This window also provides the main interface with the analyst. An example of this main window is presented in Figure 2. The **X-Y** plot, located on the left side of the window,

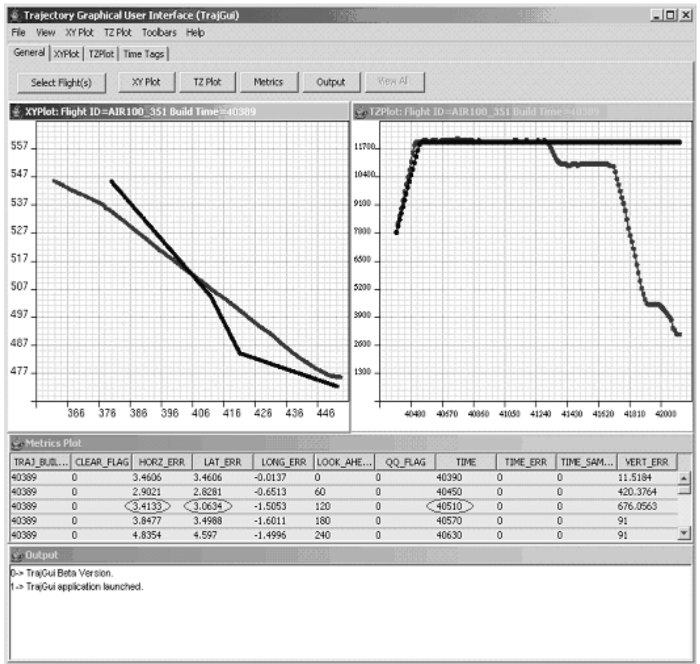

Figure 2: Main TrajectoryGUI Window

presents the positional data for the flight in a horizontal plane. It is a uniform scaled graph with units in nautical miles (nm). The positive X-axis represents east and the positive Y-axis represents north. The **T-Z** plot, located on the right side, presents the altitude data for the flight. The vertical units are feet and represent altitude with 0 being sea level. The horizontal units are seconds with the left most position of the graph denoting the beginning of the flight. The **Metrics** data is presented in a table located in below the two plots. The table is filled with the accuracy measurement data selected from the database. At the bottom of the main window is a text area that is used to report information such as operational results, program mode, and status, and errors. At the top of this window an array of functions are available to the analyst that can be used to probe the plots.

The analyst can manipulate the **X-Y** plot by using the *Re-center*, *Zoom In*, *Zoom Out*, *Offset Trajectory*, *Move Legend*, and *Reset* functions. For example, in Figure 2, the position data for the flight is shown in a viewing area of 10,000 square nautical miles. The analyst can re-center the plot by positioning the mouse at any position on the plot and clicking the left mouse button. The plot will be redrawn with the clicked position becoming the center of the plot viewing area. Figure 3 shows the result of re-centering to the area around the beginning of the flight data and zooming in to a plot view area of 100 square nautical miles. The offset functionality is useful if the trajectory lies near or directly on top of the flight's actual path. If this is the case, the analyst can offset the trajectory creating separation between the data. The analyst uses the *Move Legend* function when the legend is located in an area where data is displayed. Resetting a plot allows the analyst to return to the initial viewing area of the plot.

Other functions available to the analyst include: *Get Position*, *Get Distance*, *Clear Position*, *Clear Distance*, *Clear All Positions*, and *Clear All Distances*. When *Get Position* is activated, the analyst can click near any node, a position stamp will be applied to the closest node to the clicked location and that node's precise X and Y values will display in the text area. Examples of position stamps are shown in Figure 3 at the beginning of each path. When *Get Distance* is activated, the analyst can click near two nodes to find the exact distance between the nodes. These two nodes are usually time-coincident points. After the two nodes have been selected, a distance line will be drawn connecting the two nodes and the exact distance will be computed and displayed in the text area. When clearing positions and distances the analyst simply clicks near the item and the item is removed. Clearing all functionalities removes all of the particular items selected.

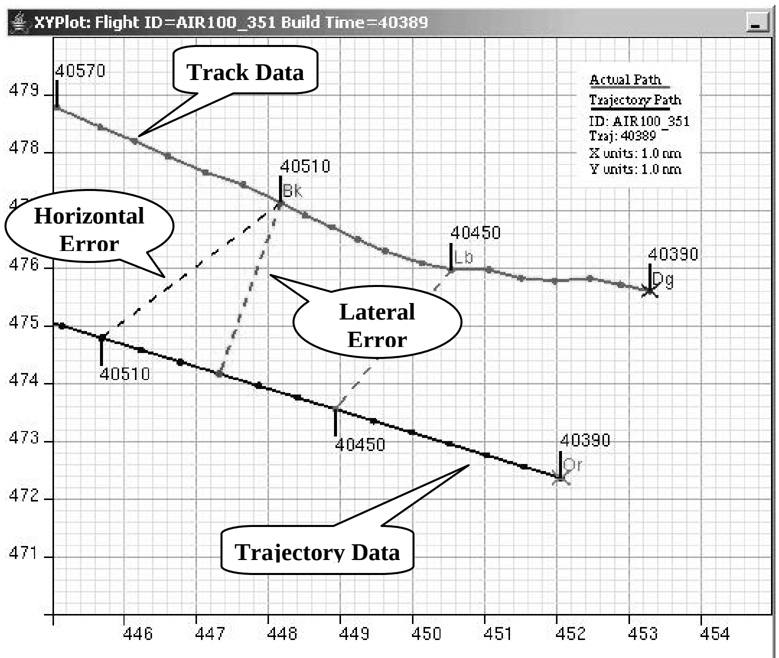

Figure 3: Manipulated X-Y Plot

Another important feature in the **X-Y** plot is the labeling of time tags, which allow the analyst to visualize the accuracy of a trajectory prediction at a specific time. The time tags are applied starting at some time point that is common in the track data and trajectory data. The time tags are based on two features: time tag frequency *f*, which is an integer value representing one time tag per *f* nodes, and offset *o*, which is an integer value representing *o* nodes from the first common time point. The analyst is given the capability to vary both the frequency and the offset value. In addition, time tags can be applied one at a time using the *Get Time Tag* function. These types of time tags are referred to as manually applied time tags and are activated when the analyst clicks on node to append time tag. Additionally, these manual time tags can be cleared similar to the position stamps and distance lines. After time tags have been applied, the time tags may overwrite viewing areas in the plot. As a result, the ability to rotate time tags is also given to the analyst.

The **T-Z** plot window displays the T and Z coordinates of the flight's track data and the flight's trajectory data in an interactive coordinate system. Many of the same features exist for the **T-Z** plot, but they are used independently within the plots. Within the **T-Z** plot, the analyst can *Re-cente*r, *Zoom In*, *Zoom Out*, *Offset Trajectory*, and *Reset*. In addition, the analyst can use the functions *Get Position*, *Clear Position*, and *Clear All Positions*. Like the **X-Y** plot, the legend can conceal valuable information at times, while the analyst can take advantage of the *Move Legend* function.

Additional capabilities of TrajectoryGUI include:

- an image export of the **X-Y** and **T-Z** plots for the use in presentations and reports,
- a file export of the **Metrics** data in a comma delimited file for import into other application programs,
- an interface that provides the ability to select and deselect the metrics that will be displayed in the **Metrics** table, and
- a configuration file for maintaining lists of:
    - accessible databases along with their connection parameters,
    - the default metric fields to be included in the **Metrics** table, and
    - the desired colors for display of the actual flight path and of the trajectory path.

Analysis Applications

Figure 3 demonstrates how TrajectoryGUI is useful in measuring the accuracy of trajectory predictions such as those presented in Reference [6]. In this figure, the actual flight path (track data) is located above the trajectory path (data) to the bottom. In this scenario, the **Metrics** table, shown in Figure 2, provides the analyst with the horizontal error and lateral error at time 40510. By applying time tags and using the Get Distance function, the analyst can verify that the distance between the track data and trajectory data at time 40510 is equal to the horizontal error located in the metric. When the distance line is drawn between the two nodes, a distance of 3.413 nm is displayed in application's text output area. This distance affirms that the horizontal error of 3.413 nm at time 40510 is accurate. Furthermore, the analyst can confirm the distance of the normal line connecting the track data at time 40510 to some point on the trajectory path. This distance is known as the lateral error. When the distance line is displayed, the analyst can refer to the text output area and see the distance value of the normal line. Using this distance the analyst can then reach a conclusion on the accuracy of the lateral error.

TrajectoryGUI has also proved its utility in a recent study that compared ARTCC track data with comparable position data provided by the Global Positioning Satellite system [7].

Future Work

As seen in Figure 1 it is planned that data for more than one flight will be simultaneously displayed on a single **X-Y** and **T-Z** plot. These flights may be from the same or from different scenario cases. In addition, additional plots will be added to broaden the analyst's ability to study the accuracy of trajectory predictions. These plots include the following measurements versus time: heading, speed, horizontal error, vertical error, and lateral error. It is anticipated that the CPAT analysts will continue to use this application as these enhancements are implemented.

Future work also exists in the development of a new graphical application that can animate one, two, or more flights. The application will read from the same type of relational database tables that TrajectoryGUI reads currently. This tool is different because it will not be plotting charts, but animating a flight's motion. The animation that is generated will be saved to an MPEG-[vi] or AVI-[vii] formatted file. This file will be used for import to other tools and for archiving purposes. The data to animate is effectively the path of one or more aircraft, displaying their spatial/time relationship to each other (separations and geometries of encounters).

Acknowledgments

The authors acknowledge SEGV alumni Lan Lin, Thomas Snyder, and Robert Zakreski for their work on TrajectoryGUI during the Spring semester of 2005.

References

[1] Cale, Mary Lee, Paglione, Mike M., Ryan, Dr. Hollis F., Timoteo, Dominic, and Oaks, Robert D., "User Request Evaluation Tool (URET) Conflict Prediction Accuracy Report," DOT/FAA/CT-TN98/8, April, 1998.

[2] Paglione, Mike, Dr. Hollis F. Ryan, Robert D. Oaks, J. Scott Summerill, Mary Lee Cale (1999), "Trajectory Prediction Accuracy Report User Request Evaluation Tool (URET)/Center-TRACON Automation System (CTAS)," DOT/FAA/CT_TN99/10, WJHTC/ACT-250.

[3] Paglione, Mike M., Lori Charles, Shurong Liu, Robert D. Oaks, Hollis F. Ryan, Ph.D, J. Scott Summerill (2002), "User Request Evaluation Tool (URET) Conflict Probe Sensitivity to Weather Forecast Errors," Draft Report, October 2002.

[4] Paglione, M. M., Singhal, S. K., Oaks, R. D., Liu, S., "Advances in the Federal Aviation Administration Testing of the User Request Evaluation Tool Using an Off the Shelf Animation Package," Air Traffic Control Association 47th Annual Conference Proceedings, November 2002.

[5] Paglione, M. M., Singhal, S. K., Oaks, R. D., Liu, S., "Using both Graphical Software and Statistical Metrics for the Evaluation of a Decision Support Tool," Air Traffic Control Association 48th Annual Conference Proceedings, October 2003.

[6] Cale, Mary Lee, Liu, Shurong, Oaks, Robert D., Paglione, Mike, Ryan, Dr. Hollis F, Summerill, Scott, "A Generic Sampling Technique for Measuring Aircraft Trajectory Prediction Accuracy," 4th USA/EUROPE Air Traffic Management R&D Seminar, December 2001.

[7] Ryan, Hollis F., Paglione, M. M., "Comparison of Host Radar Positions to Global Positioning Satellite Positions," 24th Digital Avionics Systems Conference, October 2005.

---

[i] Proof Animation is a product developed by Wolverine Software. For further information see http://www.wolverinesoftware.com.

[ii] For further information about the National Association for Equal Opportunity in Higher Education see http//www.nafeo.org.

[iii] The Java Database Connectivity (JDBC) is a Java API that enables Java applications to execute Structured Query Language (SQL) statements providing database connectivity with a wide variety of SQL-compliant databases. For further information see http://java.sun.com/products/jdbc/.

[iv] JOGL is short for Java bindings for OpenGL. The JOGL Project hosts a reference implementation for OpenGL API, and is designed to provide hardware-supported 3D graphics written in Java. It is part of a suite of open-source technologies initiated by the Game Technology Group at Sun Microsystems. For further information about the JOGL API Project see https://jogl.dev.java.net.

[v] OpenGL is an open standard for developing portable, interactive 2D and 3D graphics applications that is guided by the OpenGL Architecture Review Board. For further information see http://www.opengl.org.

[vi] MPEG refers to a family of digital video compression standards and file formats developed by the Moving Picture Experts Group; for further information see http://www.chiariglione.org/mpeg/.

[vii] AVI is short for Audio Video Interleave and is a file format for storing video and audio information developed by Microsoft Corporation.